

Accelerated Collection of Sensor Data by Mobility-enabled Topology Ranks

Constantinos Marios Angelopoulos^{a,b,*}, Sotiris Nikolettseas^{a,b}

^a*University of Patras, Greece*

^b*Research Academic Computer Technology Institute (RACTI), Greece*

Abstract

We study the problem of fast and energy-efficient data collection of sensory data using a mobile sink, in wireless sensor networks in which both the sensors and the sink move. Motivated by relevant applications, we focus on dynamic sensory mobility and heterogeneous sensor placement. Our approach basically suggests to exploit the sensor motion to adaptively propagate information based on local conditions (such as high placement concentrations), so that the sink gradually "learns" the network and accordingly optimizes its motion. Compared to relevant solutions in the state of the art (such as the blind random walk, biased walks, and even optimized deterministic sink mobility), our method significantly reduces latency (the improvement ranges from 40% for uniform placements, to 800% for heterogeneous ones), while also improving the success rate and keeping the energy dissipation at very satisfactory levels.

Keywords: Wireless Sensor Networks, Mobility, Data Collection

1. Modelling Assumptions

We study wireless sensor networks in which mobile sensor nodes are deployed over an area of interest and a mobile sink is responsible for collecting data. *Both* the sensors and the sink are assumed to be *mobile* and equipped with localization and navigational hardware. Motivated by realistic scenarios

*Corresponding author

Email addresses: aggeloko@ceid.upatras.gr (Constantinos Marios Angelopoulos), nikole@cti.gr (Sotiris Nikolettseas)

and applications, we focus on heterogeneous sensor placement distributions. Also, we examine diverse mobility dynamics for the sensor motion. We try to solve the sensory data collection problem (e.g. how the mobile sink can collect data produced at the sensors), with an emphasis on minimizing latency (while keeping the success rate and energy efficiency very high). A preliminary version of this work has appeared in IEEE GLOBECOM 2009 conference (1).

1.1. Sensor Placement and Capabilities

Sensor placement involves a $D \times D$ plane *network area*. Let n be the total number of sensors deployed. Let d be the *density* of sensors in that area (measured in numbers of sensors per m^2). Sensor devices are equipped with a set of hardware monitors that can measure environmental conditions of interest. Each device has a broadcast (digital radio) beacon mode of fixed transmission range R , and is powered by a battery. Also a sensor is equipped with a general purpose storage memory (e.g. FLASH) of small (constant) size C .

Let E_i be the initial available energy supplies of sensor i . At any given time, each sensor can be in one of three different modes, regarding the energy consumption: (a) transmission of a message, (b) reception of a message and (c) sensing of events. In our model, for the case of transmitting and receiving a message, we assume that the radio module dissipates an amount of energy proportional to the message's size. To transmit a k -bit message, the radio expends $E_T(k) = \epsilon_{trans} \cdot k$ and to receive a k -bit message, the radio expends $E_R(k) = \epsilon_{recv} \cdot k$ where $\epsilon_{trans}, \epsilon_{recv}$ are constants that depend on the radio module and the transmission range R of the sensors. For the idle state, we assume that the energy consumed for the circuitry is constant for each time unit and equals E_{idle} . Finally, the *sink* is a special, very powerful node, representing a control centre where data should be propagated to.

The *sink* is deployed in the *centre* of the network area. In the heterogeneous topology scenario, the placement distribution of sensor motes includes high density areas (called "pockets"), corresponding to hot-spots in the network. Pockets may not necessarily have the same size or density. In example, for two pockets, say A and B, pocket B may have twice as high density as pocket A. Also, the position of the pockets changes with time; this dynamics make the scenario more general and capture realistic situations in relevant applications. The rest of the network area has a uniform random placement

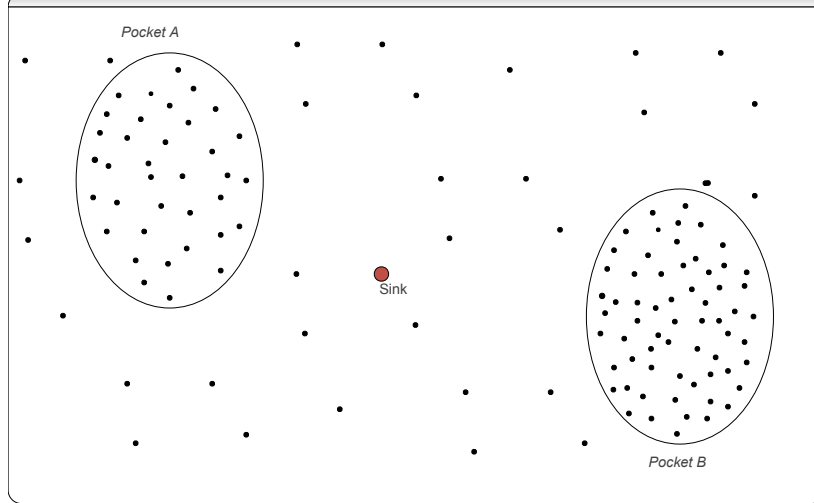


Figure 1: Heterogeneous sensor placement

(see figure 1). In the uniform scenario, the sensor distribution is uniform all over the network.

The generation rate of events is assumed to be constant ($\lambda = 0.05msg/sec$) and same for each sensor. This way, high density areas imply high sensory data generation. Also, we assume that all sensors have a common and relative (not absolute) coordination system measuring distances and time.

1.2. Modelling Dynamic Sensory Mobility

In most real world scenarios most nodes will move in many different and diverse ways. For example, a sensor attached on a vehicle will move fast on a trajectory that consists of a consecutive set of line segments. On the other hand, a pedestrian will tend to move slower over local trajectories with more curves. During these types of movement small variations of speed are usual. Also, a node will most likely change the type of movement it follows after some time varying not only the average speed but also the type of trajectory it follows. For example, consider a person working in a university campus; for long periods of time she moves slowly in a confined space (e.g. 10 by 10 meters) as she goes about her work in the office. At some point, the person may start walking faster towards a specific direction as she goes to the next building where she continues her work reversioning to the previous type of movement. These examples demonstrate the diversity and variability

that may arise in networks of mobile sensors. Modelling real life movement patterns is a subject of active research. Clearly simplistic mobility patterns, such as random walk or random waypoint alone, can not accurately capture the heterogeneous mobility characteristics we described before. Here we try to mimic, in a coarse way, several main types of movements inspired from the above observations. Using well defined mobility models, below, we define a few characteristic mobility roles that are used to construct more complex mobility behaviours.

Working movement. We parametrize a version of random walk (5) to achieve slow, located movement. We define the mobility function M_{work} with parameters $[0.5, 1.5]m/sec$ for choosing speed and by setting the movement distance towards a direction to be small, $[1, 5]m$.

Walking movement. We use a variation of the Boundless Area mobility model (5) to define M_{walk} , which is more rapid and less local than M_{work} . In this model, at each time step, a random variation of speed and direction are chosen and the next position is calculated by applying the new direction to the current coordinates. When the node reaches the boundaries of the network area we force it to reflect, i.e. take a left turn of 45° . We bound the speed to vary between $[1, 2]m/s$, we set the time step $\Delta t = 2s$; at each time step we allow the speed to vary by $\Delta v = 0.25m/s$ and the direction to vary by $\Delta a = 30^\circ$.

Bicycle ride. This type of movement is similar to the walking movement except that the speed is usually greater and there are less direction changes M_{bic} . Again we use our variation of the Boundless Area (5) mobility model; we bound the speed between $[3, 6]m/s$ ($10.8 - 21.6km/h$), we set $\Delta t = 3s$, $\Delta v = 0.5m/s$ and $\Delta \alpha = 30^\circ$.

Vehicular movement. Vehicular movement M_{veh} (5) is the faster of all; we use the Probabilistic Random Walk. In this mobility model, nodes move only towards predefined directions north, north east, east etc. We vary the speed between $[5.55, 10]m/s$ ($20 - 36km/h$).

Mobility transitions. Assigning a mobility role is enough to diversify the mobility levels of the nodes. However, in realistic scenarios nodes will change mobility roles. To model such dynamic mobility, we use a state transition diagram to change between mobility models. Each state of the diagram corresponds to a mobility role as defined above. From each state a set of outgoing edges to one or more of the other states exist; each edge is associated with a probability of transition. Also, there is an outgoing edge that returns to the same state. The sum of all outgoing edges from a

state is equal to 1. While on a state the node follows the mobility defined by the corresponding mobility model. As soon as a new position needs to be selected a probabilistic experiment is performed to choose a new state according to the state transition diagram, then the corresponding mobility function is invoked to select the position and speed of the node. We also define a special state called the stop state in which the node remains still for a small period of time (see figure 2).

Sink Mobility. The sink moves following a high level mobility function (according to the protocol used) which we symbolize by M . If p_t is the position of the sink in a given moment then $M(p_t)$ will return a new position p_{t+1} towards which the sink should move. This defines a trajectory for the sink as a series of points $p_0, p_1 = M(p_0), p_2 = M(p_1), \dots, p_t = M(p_{t-1})$. Also, the function M defines the speed $s_t = M(s_{t-1})$ by which the sink moves from position p_{t-1} to position p_t .

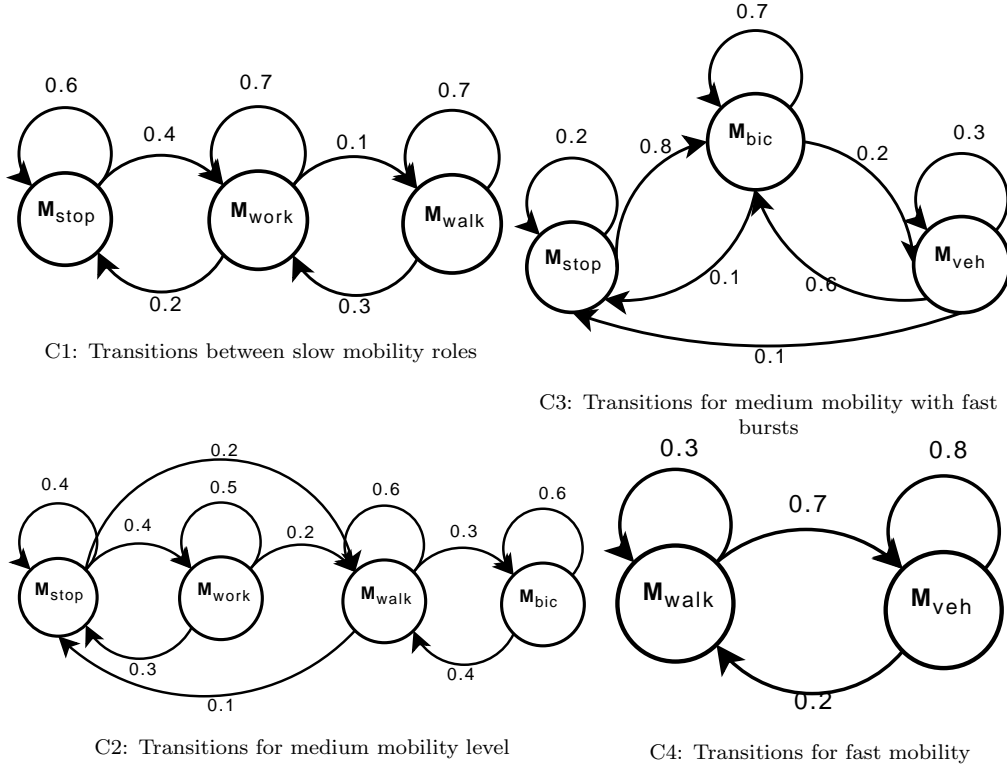


Figure 2: Mobility Transitions Diagrams

2. Related Work

In mobile settings, the protocols and findings of previous research on static WSN's can not be (at least directly) applied. Efficient solutions in the state of the art become inefficient or even inoperable. Even well studied algorithms need to be redesigned; as an example, in (3) authors propose a leader election algorithm suitable for mobile networks. Also, in (2) authors propose a mobility-aware routing protocol, using zone-based information and a cluster-like communication protocol. Additionally, new problems arise due to the high dynamics, e.g. maintaining system integrity becomes more difficult (4).

Our approach is one of the first few that considers a sensor network where both the sensors and the sink is mobile. For networks of mobile sinks, (7) investigates the network lifetime when multiple mobile sinks are periodically repositioned with respect to the energy consumption. Authors propose an integer linear program to determine the new locations and a flow-based routing protocol. (10) proposes a routing scheme where a single sink stops at certain anchor positions while collecting data. The sink samples the global power consumption while in an anchor point and determines the optimal visiting time.

For networks of mobile sensors, (9) suggests exploiting the sensor motion to adaptively disseminate data, e.g. propagate redundant data when mobility is low while propagating less data in the presence of high mobility; in contrast to our approach the sink is assumed to be static, and data travels many hops towards the sink. Considering sensor mobility, (8) presents a case study of mobile sensor networks designed for wildlife position tracking. Authors assume varying mobility and propagate data to the node most likely to meet the sink, based on previous history.

In this work, we compare our method to the following three relevant state of the art approaches:

2.1. *Blind Random Walk*

In this scenario the sink is simply moving according to a random walk process and serves any sensors that it may reach. It requires zero knowledge about the deployment of sensors and makes almost no assumption about the network, but its latency is high because of unnecessary visit overlaps and complete ignorance of network topology. This protocol in full mobility schemes is expected to present high latency as the sink does not consider the

movement of sensors. Somehow, the blind random walk represents an upper bound on latency, while its energy dissipation is very low.

2.2. Locally Adaptive Random Walk

In this protocol the sink tries to optimize its route based on local density. The sink traverses the network area following the edges of an overlaying grid. The network is divided into square $D \times D$ regions, where D is one-hop distance. Sink decides towards which of the four directions is going to follow based on the local densities of the four neighbouring regions. The decision is taken at random by assigning a probability to each direction. This probability is defined as $p_i = \frac{d_i}{d}$, $i = 1, 2, 3, 4$ where d_i is local density of region i and d is the overall density of the network.

In this protocol the sink needs to have information about the network, like the total number of sensors deployed and the size of the network area, in order to calculate d . Furthermore, it is a myopic protocol as the sink makes its decision based on nearby areas information only. Therefore, it may be attracted by nearby densities while further in the network larger concentrations of sensors may exist. For these reasons our proposed protocol is expected to perform better in terms of latency.

2.3. Optimized Deterministic Motion

In optimized deterministic motion the sink's route inside the network area is predetermined. The sink sweeps the entire area in a way that no overlaps occur. This way also guarantees that eventually all sensors are going to be reached. However, if high density areas are in distance from the sink it will take a lot of time to reach them. Even then, the sink will have to travel through the entire network area before visiting them again. So, it is expected to present higher latency compared to our proposal.

3. Our Approach

Our approach tries to exploit sensor movement in order to inform the sink about the network topology. Sensors, by carrying lightweight time-stamped information about local densities across the network, can "guide" the sink towards dense areas. This way, the sink can optimize its route and serve more sensors and higher data traffic in less time, with very limited overhead and very low energy dissipation. We below describe the basic components of our method.

3.1. Carrying and Ranking Local Topology Information

Let d_{local} be the local density of a given sub-region of the network area. At the beginning, each sensor gets informed about d_{local} corresponding to its region, by broadcasting a *hello* message and counting the sensors that respond to it. Clearly, this function is not energy efficient if every node computes d_{local} and responds to every *hello* message it receives (complexity $O(n^2)$, where n is the total number of sensors). A mechanism for reducing the number of broadcasts is described.

Intuitively, we would like fast sensors to have accurate topological information, because they can carry this information further in the network. On the other hand, we could allow slow sensors to have less accurate information as their dislocation is small. Let v_i be the speed of sensor i and let v_{max} be the top speed of sensors in the network. Then, sensor i broadcasts a *hello* message with probability $p_i = \frac{v_i}{v_{max}}$. This way, fast sensors, almost certainly, will measure d_{local} and slow sensors will measure d_{local} with low probability. When a sensor is measuring d_{local} along with the *hello* messages, it sends its speed as well. Let this speed be denoted as $sender_{sp}$. When a sensor receives a *hello* message it decides if it is going to respond or not via

$$a_i = \frac{sender_{sp}}{receiver_{sp}}$$

where $receiver_{sp}$ is the speed of the receiving sensor. If $a_i > 1$ then the receiver responds, as we have a faster sensor asking a slower one. Else the receiver decides with probability a_i if it will respond. Note that the slower is a sensor, the smaller is the probability to respond.

When sensors start moving, each one carries this local topology information. As said, our goal is to gradually inform the sink about the local distribution of sensors in different regions of the network. If the sink is aware of a high concentration of sensors in a sub-region (thus, high local density and high data traffic), it will try to reach this sub-region as quickly as possible. So, each sensor carries information about the sub-region corresponding to its d_{local} , as it moves in the network.

Let P be the coordinates of a given position inside the network area. This type of information can easily be acquired via navigation hardware. Intuitively, the value of information carried by d_{local} degrades over distance. The further a region is, the more time will take for the sink to reach it, and given the dynamic mobility of sensors, the less likely it is for this information to be

”correct” by the time the sink gets there. The same degradation of quality occurs over time as well. The older d_{local} is, the less accurate (i.e. obsolete) the information it carries. Let T be a denotation of time, i.e. when d_{local} was obtained. Again, this information can be acquired from a synchronised clock.

By now, the necessity of a ranking function denoting the quality and importance of carried information should be obvious. Let

$$R = \frac{d_{local}^2}{\Delta P \Delta T} \quad (1)$$

where ΔP is distance change (dislocation) and ΔT a time interval, be the ranking function, that ranks information of d_{local} with respect to the distance from origin and the time it was measured. Based on this function, the sink decides towards which direction it will move via a mechanism described in following sections.

3.2. Updating Ranked Information

Previously, we have seen that each sensor can carry local topology information for a sub-region of the network area simply by taking measurements for local density and corresponding position and time. Then, it estimates the importance of these information through the ranking function $R = f(d_{local}, \Delta P, \Delta T)$. As the quality of carried information degrades, each sensor should try to replace it with new information of better quality. So, periodically, each sensor gets informed about d_{local} and marks current position and time. Because of memory restrictions, information of degraded quality is discarded, allowing to use minimal space in memory.

Consider a single sensor. The update process begins with a first set of measurements. Let them be denoted as d_{local}, P and T . The sensor is moving in the network and after a period of time (that is defined by the protocol) a second set of measurements is taken. Let d'_{local}, P' and T' be the second set. When a third set of measurements is to be taken, sensor must decide which of the two old sets it should discard. This can be easily decided via the ranking function R . Let $P_{current}$ and $T_{current}$ be the current values for position and time. Let

$$r = \frac{d_{local}}{||P_{current} - P|| (T_{current} - T)}$$

be the rank corresponding to the initial set and

$$r' = \frac{d'_{local}}{||P_{current} - P'||(T_{current} - T')}$$

the rank for the second. Then, the set of measurements corresponding to $\min\{r, r'\}$ is to be discarded and replaced by the new measurements. Continuously performing this process guarantees that each sensor will carry the best ranked information available.

Note that by following the described scheme, each sensor needs very small, constant space in memory (enough to store two sets of information only). Our method can be extended to carry a higher number of triplets (say β), that clearly introduces a performance vs cost trade-off, since a large β would provide a more accurate topology information, but at the same time more memory would be needed.

Furthermore, information concerning regions with high density values (i.e. pockets) will achieve high rankings that will be carried by the sensor for a longer time period and eventually traverse a longer distance inside the network area. Following, we define the process followed by the sink to exploit this ranked information in order to optimize its route, eventually reducing latency in a power-efficient way.

3.3. Exploiting Ranked Information to Optimize Sink Motion

The sink starts with no information regarding the distribution of sensors over the network. It moves in a random direction, waiting to encounter sensors. When a sensor gets inside the communication range of the sink, along with sensory data, is asked to deliver its ranked information regarding network topology. That is, the stored triplet (d_{local}, P, T) that the very moment of communication with the sink achieves the higher value for function R . However, a sensor responds to this requiry only if it has data to deliver. So, areas of no interest or areas that have been recently served by the sink are ignored. This way the sink doesn't get "trapped" inside dense areas (i.e. pockets), but leaves them as soon as all data is collected. Sink collects topology information for a short period of time (round) from any sensor that it may contact during this period. Suppose that the sink has collected m triplets of information, each one having a rank. The sink will move along the direct line defined by current position and the position corresponding to the best-ranking triplet. If there is no topological information

available, the sink simply follows the blind random walk until it contacts at least one sensor. Having moved towards this direction for a distance, say of 3 hops, the sink starts, again, to collect topology information from nearby sensors. This way, it can re-evaluate its route. It is expected that the closer the sink gets to a subregion of high density, the more sensors will urge it to move towards this region. If for some reason (i.e. mobility, failures, etc) the local density of that region drops, the sink will collect high ranks regarding different locations, thus redefining its route.

3.4. *Some Variations*

3.4.1. *Probabilistic Variation*

The first variation of our protocol lies on the way the sink chooses its next direction based on received topological information. Suppose that the sink has collected m triplets of information. Let $r = \{r_i\}, i = 1, 2, \dots, m$ be the set of ranks corresponding to these triplets. The sink assigns a probability

$$p_i = \frac{r_i}{\sum_j r_j}, j = 1, 2, \dots, m$$

to each member r_i of r , and chooses one based on this probability. Then, it decides to move along the direct line defined by current position and the P value of the chosen triplet. A probabilistic choice has the advantage of balancing over the network area the number of visits by the sink.

The reason why the sink decides probabilistically about its route, in this variation, is because we wish to globally balance this decision. If the sink simply chose the highest r_i , areas with low density would be ruled out from sink's route and so, would not be served. Also, probabilistic decision making is a fair process for choosing among almost equal values of the R function. However, there is a slight chance that this probabilistic decision-making process will increase latency, that is the time interval since a data message is born until it is delivered to the sink. This might happen due to the fact that when the sink is notified about a grand concentration of sensors, there is a (small) probability that it will ignore it.

3.4.2. *Multiple Sinks*

Finally, the protocol can be interestingly extended to multiple sinks. In this case challenges concerning inter-sink coordination arise. Having more

than one sinks allows multiple regions of the network to be served simultaneously. However, the ranking function $R = f(d_{local}, \Delta P, \Delta T)$ drives all sinks to move towards high density areas. For these reasons we introduce the following coordination process:

Let s be the number of sinks deployed in the network area. Let every sink have a queue Q of size $s - 1$. When each sink starts getting topological information from sensors (triplets of (d_{local}, P, T)) it inserts them into the Q in descending order. If two sinks get inside the communication range of each other, they compare their topological information. If they are referring to different network regions they continue with their route. If they are routing towards the same or neighboring regions then they compare the quality of their information via the R function. The sink with the higher information ranking continues with its route. The sink with the lower information ranking changes its route towards the corresponding region of the second triplet in its queue. The worst case for a sink is to contact with every other sink in the network at the same time. For this reason the size of queue is $s - 1$. This coordination process benefits the sink(s) with the higher ranking triplets. In fact, it allows this sink to move unobstructed towards its destination, therefore adding delay (due to direction sifting) to sinks carrying topological information of lower quality.

Another possible coordination process could include a mutually avoiding scheme for the sinks. In this scheme, when two or more sinks are approximating each other, they repulse each other with a “force” that is reversely proportional to their distance. It is expected that following this process would eventually lead the sinks to separate the network area into “territories” and therefore to faster data collection. All these variations are subjects for future work and are planned to be thoroughly investigated through detailed experiments.

4. Performance Evaluation

We implement our protocols in the ns-2 simulation platform version 2.33, using the TRAILS toolkit (6), which simplifies the implementation and simulation of complex mobility scenarios. We have studied two representative scenarios, one including heterogeneous placements for sensors, and a second one including uniform placements. We set the network area to be $500 \times 500 m^2$, we always position the sink S at (250, 250), the center of the network, except for the Optimized Deterministic Motion in which we position S at (15,15)

in order to be fair with the protocol. Otherwise, at the beginning, it would visit two times the same half of the network, and it would take twice such time to visit the other half. For each protocol we run 5 sets of experiments, for 50,100,150,200 and 250 sensors. Each set consists of 20 iterations and computes the mean value of each metric.

For the heterogeneous placement scenario, the sensor placement consists of two pockets A and B, each one of $15 \times 30 m^2$. In pocket A 52% of all sensor population is deployed, in pocket B 27% and the remaining sensors are deployed uniformly in the rest of the network area. Hence, pocket A has twice more density than pocket B and each pocket has significantly higher density than the rest of the network area. Furthermore, the coordinates of the pockets change every 1500secs of simulation time (that is, sensors gradually gather to different positions), increasing this way network heterogeneity and making the scenario more general. For the uniform placement scenario all sensors are deployed uniformly inside the entire network area.

The sink has significant energy resources (100Joules) and has a constant speed of $8 \frac{m}{sec}$. Each sensor commences with 5 Joules of energy. The sink S transmits beacon messages at a steady pace of $\lambda_{Beacon} = 1$, that is a beacon message per second, and asks for topological information every 3secs. In our application scenario we assume that all sensor nodes record an instance of the environmental conditions producing a fixed number of data messages set to 50. The time interval between two successive messages produced at a node i is not constant, messages are produced at random intervals. However, on the average new messages are produced at rate $\lambda_i = 0,05$ messages/second. Thus, the data generation phase lasts for about 1000sec, we simulate the network for 5000sec, in order to collect delayed data. The data is generated in packets of 36 bytes while the size of a beacon message is 24 bytes. The transmission range of both nodes and sink is set to $R = 15m$. The characteristics of the radio module, i.e. the values of e_{trans} , e_{recv} and E_{idle} , were set to match as closely as possible the specifications of the mica mote platform.

4.1. Node movement

We assign different mobility roles to the nodes of the network. We examined cases where the mobility of the nodes changes during the simulation using the mobility transition graphs defined earlier. In particular, we assign C_1 to 25% of the nodes, C_2 to another 25%, C_3 to another 25% and C_4 to the remaining 25% of the nodes, with speed having a mean value of $0.8 \frac{m}{sec}$, $3 \frac{m}{sec}$, $9 \frac{m}{sec}$ and $18 \frac{m}{sec}$ accordingly.

4.2. Metrics

Conducting these experiments, we measure several metrics that depict the performance of the protocols. We call success rate the percentage of data messages that were received by the sink over the total number of generated messages. We measure the energy consumed at the sensor network due to communication, as the average number of Joules consumed at each node. We also measure the delivery delay (latency), which is the average time interval between the creation of a message and the time when it is delivered to the sink.

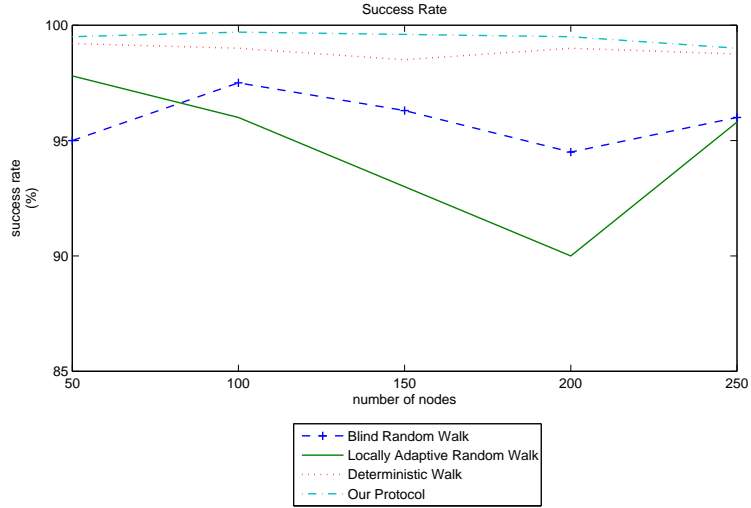


Figure 3: Success rate in heterogeneous placement scenario

4.3. Performance

We would like to point out that the statistical analysis of our findings shows high concentration around the mean. Also, overlaps concern the protocols we compare with, while ours clearly outperforms the other ones, i.e. improvement of latency in heterogeneous placements exceeds 800%. The success rate and energy dissipation are very satisfactory, similar to the other protocols (thus the overlap).

4.3.1. Heterogeneous Placement

Figure 3 depicts the success rate achieved by each protocol. Locally adaptive protocol achieves smaller ratios as it is a "myopic" protocol, thus having difficulties locating the faraway pockets. Once it locates a pocket it ignores the second one. Even worse, when pockets are reformed in a new position, this protocol again is having difficulties in finding the new pockets. All other protocols perform relevantly the same, however our protocol presents a slightly better success rate.

Figure 4 depicts the latency that each protocol presents. As expected, blind random walk and optimised deterministic walk perform more or less the same under mobile sensor networks with high dynamics and heterogeneity, with latency varying from 1000secs to 1250secs. Again, locally adaptive protocol shows an unstable performance. Our proposed protocol behaves extremely well by outperforming the rest of the protocols by even 800%, because sensors with high mobility inform the sink by carrying topological information far in the network area.

Figure 5 depicts the amount of energy dissipated in the network in Joules per sensor. The three protocols we compare with dissipate approximately

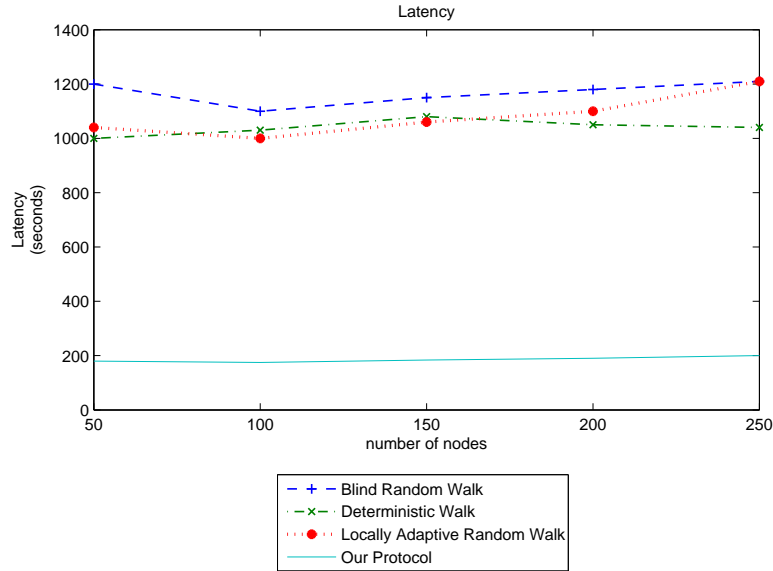


Figure 4: Latency in heterogeneous placement scenario

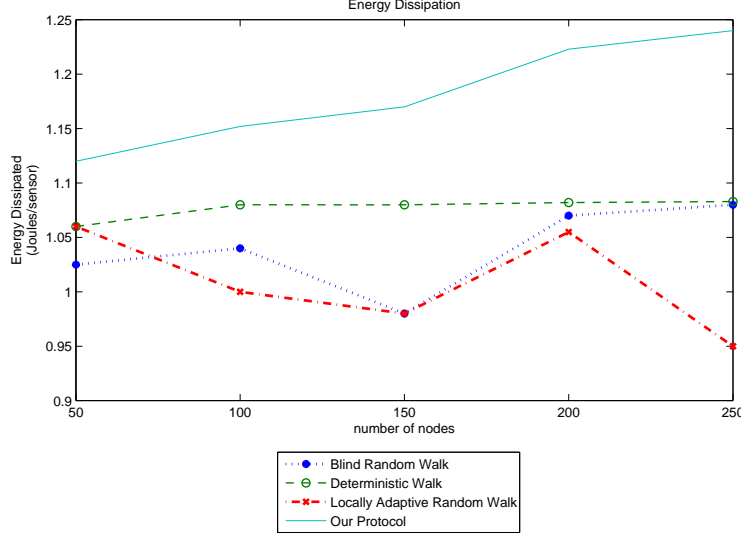


Figure 5: Energy Dissipation in heterogeneous placement scenario

the same amount of energy per sensor. This happens because in all these protocols sensors have a passive role simply delivering data when they contact the sink. Thus, the energy dissipated is depending on the data messages delivered. Our protocol presents a slightly higher dissipation, because sensors collect topological information as well.

4.3.2. Uniform Placement

Accordingly, figure 6 depicts the success rate of each protocol in the uniform placement scenario. As expected optimized deterministic motion achieves higher values than blind random walk as it sweeps the network in an optimized way, and each region inside the network area has approximately the same density. Locally adaptive random walk presents a lower success ratio because all four directions have the same density, the choice among them is balanced and so the sink traverses the network slowly. Our protocol, disallows a given sensor to send topological information to the sink, *unless* it has sensory data to deliver. This allows the sink to not overlap (i.e. not visit) recently visited areas. Thus, our protocol performs very well in uniform placements, even outperforming optimized deterministic motion for

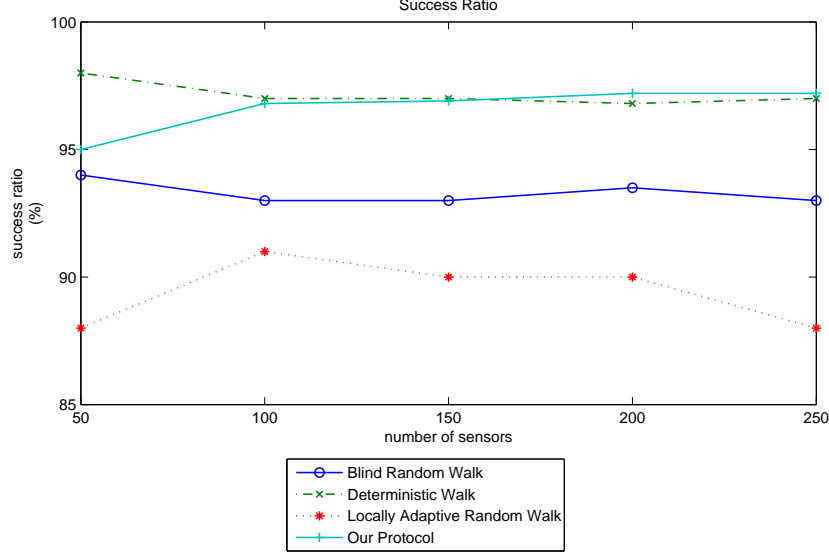


Figure 6: Success rate in uniform placement scenario

larger numbers of sensors. For the same reasons, in latency our protocol outperforms even by 40% the other ones, as is shown in figure 7.

Figure 8 depicts the dissipated energy per sensor. The amount of energy dissipated in other protocols depends on how many data will be delivered, so more effective protocols present higher energy dissipation. However, our protocol dissipates only 15% more energy on the average than other protocols, thus being energy efficient.

4.4. Conclusions and Future Work

In this work we studied sensor networks in which both the sensors and sink are mobile, and sensors move in a diverse, highly dynamic manner. Motivated by realistic scenarios and applications we focused on heterogeneous sensor placements distributions. We proposed a mobility-based topology exploration protocol in which the sink gets gradually informed about network topology by local information that mobile sensors collect and carry. This way the sink can effectively collect data produced reducing latency even by 800%, compared to relevant solutions, while keeping energy dissipation at low levels. Still, our protocol outperforms other ones even in uniform placement scenarios since the sink is able to locate even small traffic discrepancies over time

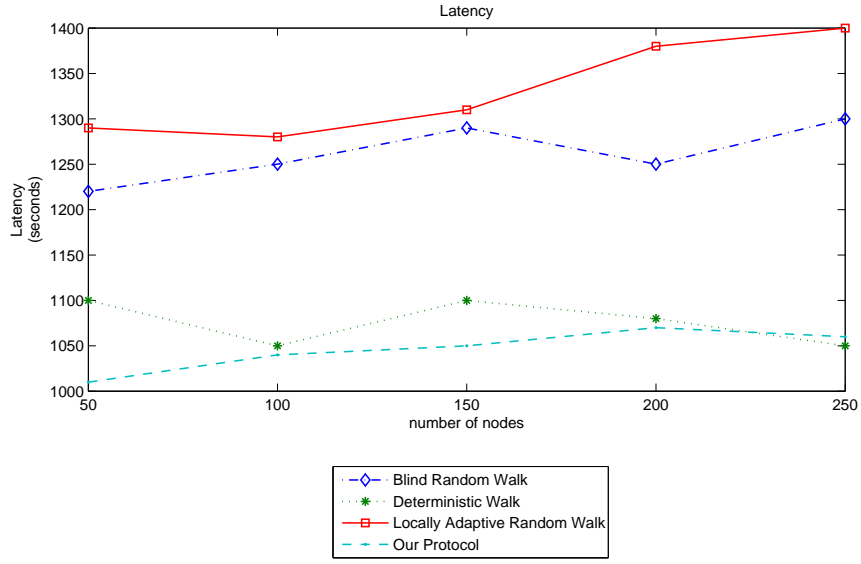


Figure 7: Latency in uniform placement scenario

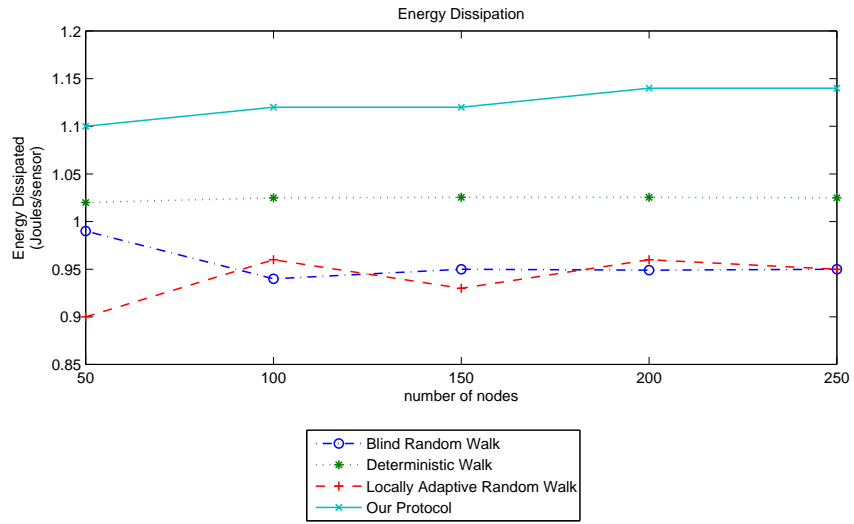


Figure 8: Energy Dissipation in uniform placement scenario

and head fast towards higher traffic regions. In future work we will study more effective methods of topological information exploitation (e.g. aggregating ranks) as well as more effective ways (optimized sampling methods) for collection of these information by sensors.

Acknowledgments

This work has been partially supported by EU/FIRE HOBNET project - STREP ICT-257466

References

- [1] Constantinos Marios Angelopoulos and S. Nikolettseas, Fast Sensory Data Collection By Mobility-based Topology Exploration, in Proceedings of the 52nd IEEE International Global Communications Conference (GLOBECOM), 2009.
- [2] L. M. Arboleda C. and N. Nasser. Cluster-based routing protocol for mobile sensor networks. In QShine, page 24, 2006.
- [3] A. Boukerche and K. Abrougui. An efficient leader election protocol for mobile networks. In International Conference on Wireless Communications and Mobile Computing (IWCMC), pages 1129-1134, New York, NY, USA, 2006. ACM.
- [4] A. Boukerche, Y. Ren, and Z. Zhang. Performance evaluation of an anonymous routing protocol using mobile agents for wireless ad hoc networks. In 32nd IEEE LCN, pages 893-900, 2007.
- [5] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad-hoc network research. Wireless Communications & Mobile Computing, pages 483-502, 2002.
- [6] I. Chatzigiannakis, A. Kinalis, G. Mylonas, S. Nikolettseas, G. Prasinos, and C. Zaroliagis, TRAILS, a toolkit for efficient, realistic and evolving models of mobility, faults and obstacles in wireless networks, 41st ACM/IEEE Annual Simulation Symposium (ANSS), 2008, pp. 23-32.

- [7] S.R. Gandham, M. Dawande, R. Prakash, and S. Venkatesan. Energy efficient schemes for wireless sensor networks with multiple mobile base stations. In IEEE GLOBECOM, volume 1, pages 377-381 Vol.1, Dec. 2003.
- [8] P. Juang, H. Oki, Y. Wang, M. Martonosi, L. Peh, and D. Rubenstein. Energy-efficient computing for wildlife tracking: Design trade-offs and early experiences with zebranet. In 10th ASPLOS, 2002.
- [9] A. Kinalis and S. Nikolettseas, "Adaptive Redundancy for Data Propagation Exploiting Dynamic Sensory Mobility", In the Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM), ACM Press, pp. 149-156, 2008. Also, in the Journal of Interconnection Networks (JOIN) 2009.
- [10] J. Luo, J. Panchard, M. Piorkowski, M. Grossglauser, and J-P. Hubaux. Mobiroute: Routing towards a mobile sink for improving lifetime in sensor networks. In 2nd IEEE DCOSS, volume 4026, pages 480-497, 2006.
- [11] O. Powell, P. Leone, and J. Rolim. Energy optimal data propagation in wireless sensor networks. Journal of Parallel Distributed Computing (JPDC), pages 302-317, 2007.